

PATENT

**C. REMARKS****Status of the Claims**

Claims 1, 3-8, 10-14, and 16-23 are currently present in the Application and stand rejected. Claims 1, 8, and 14 are independent claims. No claims have been amended, added, or cancelled in this Response.

**Claim Rejections - Alleged Anticipation Under 35 U.S.C. § 102**

Claims 1, 3-8, 10-14, and 16-23 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated, and therefore unpatentable, over U.S. Patent No. 6,327,551 to Peterson et al. (hereinafter "Peterson"). Applicants respectfully traverse the rejections.

Applicants claim a method/system/program product for developing topography based management systems. Each of Applicants' independent claims include the limitations of:

- analyzing a topography design corresponding to a topography;
- identifying one or more topography requirements based on the analysis;
- creating topography components corresponding to the identified topography requirements, wherein each of the components is adapted to interoperate with one or more operating environments, and wherein at least one of the components is a topography neutral application component that is adapted to interoperate with more than one topography; and
- storing component data in a topography data store, the component data describing one or more of the components.

The Office Action contends that each of these limitations is taught by Peterson. A review of the Peterson reference, however, reveals that Peterson does not teach or suggest Applicants' claimed limitations.

Applicants respectfully direct the Examiner's attention to MPEP § 2131, which states, in part, that in order "to anticipate a claim, the reference must teach every element of the claim." The text of MPEP § 2131 is as follows:

**2131 - TO ANTICIPATE A CLAIM, THE REFERENCE MUST TEACH EVERY ELEMENT OF THE CLAIM**

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros.*

Docket No. AUS920010638US1

Page 7 of 19  
Sweitzer, et. al. - 09/996,131

Atty Ref. No. IBM-1038

## PATENT

v. *Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). >"When a claim covers several structures or compositions, either generically or as alternatives, the claim is deemed anticipated if any of the structures or compositions within the scope of the claim is known in the prior art." *Brown v. 3M*, 265 F.3d 1349, 1351, 60 USPQ2d 1375, 1376 (Fed. Cir. 2001) (claim to a system for setting a computer clock to an offset time to address the Year 2000 (Y2K) problem, applicable to records with year date data in "at least one of two-digit, three-digit, or four-digit" representations, was held anticipated by a system that offsets year dates in only two-digit formats). See also MPEP § 2131.02.< "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim, but this is not an ipsissimis verbis test, i.e., identity of terminology is not required. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990). Note that, in some circumstances, it is permissible to use multiple references in a 35 U.S.C. 102 rejection. See MPEP § 2131.01.

As detailed below, the Peterson reference falls considerably short of the requirements set under MPEP § 2131 for rejecting Applicants' claims under 35 U.S.C. § 102. Applicants note that the text cited in Peterson to support the Office Action's rejection often has nothing to do whatsoever with Applicants' claimed limitations. Applicants note that Peterson is directed at a designing computer systems using Peterson's system design methodology, while Applicants' invention is directed at developing topography based management systems. Applicants further note, as explained in detail below, that Peterson does not teach or suggest creating components for multiple topographies, nor does Peterson teach or suggest components that are topography neutral, each as claimed by Applicants.

The Office Action contends that Peterson teaches Applicants' claimed *analyzing a topography design corresponding to a topography*, citing col. 1, lines 18-33 and col. 2, lines 23-25. In the sections cited in the Office Action, Peterson reads as follows:

Col. 1, lines 18-33 reads as follows:

The waterfall life-cycle model comprises a series of process steps, as illustrated in FIG. 1 and is an analytic approach which is sometimes referred to as such. It starts with a set of requirements and proceeds via implementation and other process steps to an operations and maintenance process. Feedback is possible between adjacent processes in the design sequence and the following processes may be included sequentially:

- System feasibility
- Software requirements
- Preliminary design
- Detailed design

## PATENT

Code and debug  
Test and pre-operation  
Operations maintenance

Col. 2, lines 23-25 reads as follows:

In today's markets it is frequently necessary to produce extremely complex custom-tailored systems with great expedition.

In the cited sections, Peterson is describing a standard, well known system design methodology and the fact that complex custom-made systems are often needed by customers. Importantly, Peterson does not teach or suggest analyzing a topography design that corresponds to a topography.

Applicants note that the Office Action may be confusing Peterson's use of "topological representations" with Applicants' claimed "topography." A review of the Peterson reference reveals that Peterson's topological representations are quite dissimilar to Applicants' term "topography."

According to a fifth aspect of the present invention there is provided a design engine, for producing a requirement specification of a product, system or service from a user model embodied in a usage specification of said product, system or service, characterised (sic) in that said design engine includes storage means for said user model, means for deriving primary user goals from said user model, storage means for storing a representation of primary user goals, means for deriving a subgoal structure from said primary goals, storage means for storing a topological representation of said subgoal structure, means for deriving service states of said product, system or service from said subgoal structure, storage means for storing representations of said service states, means for generating service tasks and service objects from said service states and the topological representation of said subgoal structure and storage means for storing representations of said service tasks and service objects, the requirement specification comprising the aggregate of the representations of subgoals, service states, service objects and service tasks.

Col. 5, lines 43-62 (emphasis added)

As can be seen from the section above, Peterson describes "topological" representation as a way of storing a subgoal structure, with a subgoal being derived from a goal of the system that is being designed using Peterson's system. Throughout Applicants' specification, Applicants describe their "topography" as being quite different from the topological representation taught by

## PATENT

Peterson. For example, in Applicants' Summary (page 5, lines 4-23), Applicants describe "topography" as follows:

[A] topography that is suitable to a particular management philosophy or particular customer requirements. The topography can be viewed as a fabric that provides an infrastructure that supports the customer's management philosophy and other requirements.

For example, in a distributed system, the topography would extend to the distributed components of the system and provide communication capabilities between processes running on the various components. In addition, the topography would address deployment mechanisms, such as interfaces between applications and users, security infrastructure, such as what control is asserted and maintained for the topography, component interaction defining how components installed on the topography interact with one another, and operation conduits that determine where and how processing is performed within the infrastructure. These same capabilities are addressed by other topographies that are developed. For example, topographies for large, centralized systems as well as small, stand-alone systems can be identified and built.

The Office Action rejects the next limitation of Applicants' independent claims (identifying one or more topography requirements based on the analysis), as being taught by Peterson at col. 2, lines 23-25, and col. 6, lines 11-13, 22-26, and 58-65. As an initial matter, because Peterson never teaches or suggests the analysis of a topography design, Peterson simply cannot teach or suggest identifying anything based on such analysis. The cited sections of Peterson read as follows:

Col. 2, lines 23-25 reads as follows:

In today's markets it is frequently necessary to produce extremely complex custom-tailored systems with great expedition.

Col. 6, lines 11-13 reads as follows:

detailed design followed by validation, 4; coding and debugging phase followed by a development test, 5;.

Col. 6, lines 22-26 reads as follows:

The design method of the present invention is built on a user centred (sic) approach which starts with a usage specification expressing the market opportunity in terms of the expected user goals, user imposed constraints, economic and technical factors and the desired performance characteristics.

## PATENT

Finally, Col. 6, lines 58-65 reads as follows:

After the market opportunity has been described, task analysis is used to breakdown or decompose the main goal into a set of sub-goals or sub-sub-goals. A functional requirement analysis of the goal tasks then identifies the service states, i.e. the states of the work system that enable users to complete their goals, the associated service tasks, i.e. the set of tasks necessary to produce the service states, and the requirements placed on these tasks.

Again, the cited sections of Peterson are directed towards a system design method that is built on a “user centered approach” with no mention of identifying topography requirements based on an analysis of a topography design. As previously explained, Peterson does not teach or suggest “analyzing a topography design” and therefore is incapable of “identifying ... topography requirements” based on such an analysis.

As Peterson does not teach or suggest “identifying ... topography requirements,” it follows that Peterson cannot teach Applicants’ next limitation of *creating topography components corresponding to the identified topography requirements, wherein each of the components is adapted to interoperate with one or more operating environments, and wherein at least one of the components is a topography neutral application component that is adapted to interoperate with more than one topography.*

The Office Action contends that Peterson teaches this limitation citing Peterson at col. 15, lines 20-41 and 60-64. The cited sections of Peterson read as follows:

Peterson, col. 15, lines 30-41:

Each category has a corresponding layer. These layers may contain reusable information, components, supply structures and entities. In this way a conceptual specification can be produced from the requirements analysis by using or reusing concepts from the conceptual layer. Similarly the functional specification is produced from the requirements specification by means of a mapping from the requirements specification using or reusing components from the functional layer (these components are functions and objects). This process also holds good for the interface and device layers. Of course, when no reusable component exists design must be performed at the lower levels.

Peterson, col. 15, lines 60-64:

Conceptual layer:

This layer contains, for example, abstract concepts such as topologies (network, star, ring) and design metaphors (copier interface, dishwasher interface, catalogue) and structures of tasks or tools (hierarchical, network).

Docket No. AUS920010638US1

Page 11 of 19  
Sweitzer, et. al. - 09/996,131

Atty Ref. No. IBM-1038

## PATENT

It is clear from the recited sections that Peterson's "components" are components used during the design process in order to create a functional specification. As Peterson teaches, and as known by those skilled in the art, various specifications are often created, including functional specifications and requirements specifications, as discussed by Peterson before creating software that will be used by the end user or customer. The "components" taught by Peterson are used in Peterson's system that aids a user in designing product specifications more rapidly (see Peterson, Abstract).

Furthermore, the above limitation found in each of Applicants' independent claims includes a further limitation of "wherein at least one of the components is a topography neutral application component that is adapted to interoperate with more than one topography." As discussed above, the "components" taught by Peterson are used in Peterson's system aids in the development of product specifications. Nowhere does Peterson teach or suggest that the components used in Peterson's system are "topography neutral" and are adapted to "interoperate with more than one topography," as claimed by Applicants.

Finally, the Office Action contends that Peterson teaches the limitation of "storing component data in a topography data store," citing col. 6 lines 11-13 and col. 5, lines 6-25. As explained above, Peterson does not teach creating components as taught and claimed by Applicants. Therefore, it follows that Peterson does not teach storing component data in a topography data store.

Peterson, col. 5, lines 6-25:

According to a sixth aspect of the invention there is provided a design engine for producing a requirement specification, for a features manager for a new service created from a plurality of known service features, from a user model embodied in a usage specification of said new service characterised (sic) in that said design engine includes storage means for said user model, means for deriving primary user goals from said user model, storage means for storing a representation of primary user goals, means for deriving a subgoal structure from said primary goals, storage means for storing a topological representation of said subgoal structure, means for deriving service states for said features manager from said subgoal structure, storage means for storing representations of said service states, means for generating service tasks and service objects from said service states and the topological representation of said subgoal structure and storage means for storing representations of said service tasks and service objects, the requirement specification comprising the aggregate of representations of subgoals, service states, service objects and service tasks.

Docket No. AUS920010638US1

Page 12 of 19  
Sweitzer, et. al. - 09/996,131

Atty Ref. No. IBM-1038

## PATENT

Peterson, col. 6, lines 11-13:

detailed design followed by validation, 4; coding and debugging phase followed by a development test, 5;

The cited sections of Peterson serve to buttress Applicants' assertion that Peterson does not teach or suggest Applicants' claimed invention of analyzing a topography design, identifying topography components based on the analysis, creating the components, and storing the component data. The data being stored by Peterson are related the goals, subgoals, and structures used in "producing a requirement specification." As previously described, Peterson's system is used to develop requirements that, in turn, might be used to build application components. Importantly, Peterson does not teach or suggest creating "topographical components" and storing information related to such components.

As outlined above, Peterson clearly does not teach or suggest each of Applicants' claimed limitations set forth in each of Applicants' independent claims as required to support the rejection of these claims. Consequently, the rejection of Applicants' independent claims under 35 U.S.C. § 102 has been overcome and such claims are allowable over the prior art of record.

The remaining claims each depend, directly or indirectly, on one of the allowable independent claims and, therefore, are allowable for at least the same reasons that the independent claims are allowable. Notwithstanding the allowability of these claims for this reason alone, in the following paragraphs Applicants discuss the allowability of the dependent claims on separate grounds.

Dependent claims 3, 10, and 16 are Markush claims specifying that at least one of the topography requirements is selected *from the group consisting of a communication framework, a deployment mechanism, a security infrastructure, and an operation conduit*. The Office Action contends that Peterson teaches this limitation, citing col. 15, lines 59-64. This section of Peterson reads as follows:

Conceptual layer:

This layer contains, for example, abstract concepts such as topologies (network, star, ring) and design metaphors (copier interface, dishwasher interface, catalogue) and structures of tasks or tools (hierarchical, network).

Docket No. AUS920010638US1

Page 13 of 19  
Sweitzer, et. al. - 09/996,131

Atty Ref. No. IBM-1038

## PATENT

This section of Peterson describes a “conceptual layer” in the design of a system. The “topologies” described by Peterson relate to standard network topologies and are not interchangeable with Applicants’ “topographies,” described in detail above. The “conceptual layer” of the design is one of the earliest phases and, again, Peterson describes “documentation” that is created during this phase. The documentation is then used to develop other specification documents. The “conceptual layer” components of Peterson, therefore, are simply not analogous nor do they teach or suggest the topography requirements claimed by Applicants in claims 3, 10, and 16.

Dependent claims 4, 11, and 17 are Markush claims specifying that *the component data includes one or more fields selected from the group consisting of a component identifier, a target platform, a development environment, a control model, a topography scale, a management style, a component dependency, a component placement, a component packaging data, a component bundling data, a component build option, and a component runtime option.*

The Office Action contends that Peterson teaches these limitations and again cites col. 15, lines 59-64 in support of the rejection. Applicants are confused as to how the same short section of Peterson can possibly be used to reject the limitations set forth in these claims which are different from the limitations set forth in claims 3, 10, and 16. Again, Peterson is teaching a “conceptual layer” and information it contains and does not teach or suggest anything to do with “component data” nor does Peterson teach or suggest any of the elements set forth in Applicants’ Markush claims.

Dependent claims 5, 12, and 18 each add the limitations of:

- saving each component in a component library;
- wherein the storing further includes writing a record in a database file, each record corresponding to a distinct component.

The Office Action contends that Peterson teaches the “saving” limitation and cites col. 16, lines 1-8 in support thereof. The Office Action also contends that Peterson teaches

## PATENT

Applicants' "wherein..." clause and cites col. 16, lines 1-8 and col. 15, lines 30-42 in support thereof.

Peterson, col. 16, lines 1-8:

The conceptual layer consists of subsystems and tool boxes. A subsystem can be regarded as a set of related tool boxes and a tool box as a set of related tools and entities (objects). A possible example is a service or system giving support to a sales organisation (sic). Some possible subsystems are the set of tools supporting various roles within the organisation (sic), other subsystems may be task related or goal related.

Peterson, col. 15, lines 30-41:

Each category has a corresponding layer. These layers may contain reusable information, components, supply structures and entities. In this way a conceptual specification can be produced from the requirements analysis by using or reusing concepts from the conceptual layer. Similarly the functional specification is produced from the requirements specification by means of a mapping from the requirements specification using or reusing components from the functional layer (these components are functions and objects). This process also holds good for the interface and device layers. Of course, when no reusable component exists design must be performed at the lower levels.

The cited sections relate to the "conceptual layer" of Peterson's system design methodology and to "categories" within each layer of the design. Importantly, neither section teaches or suggests "saving" any component, nor does either section teach or suggest using a component library, each as claimed in these dependent claims. Moreover, neither section teaches or suggests writing a record to a database file with the records corresponding to a component, also as claimed in each of these claims.

Dependent claims 6, 13, and 19 add the limitations of:

- identifying one or more client attributes corresponding to a client;
- comparing the identified client attributes to the topography components; and
- selecting one or more topography components based on the comparing.

The Office Action contends that Peterson teaches each of these claimed limitations, citing col. 15, lines 30-41 and 60-64 and col. 26, lines 43-46 as support of each of these limitations. Each of these sections of Peterson are discussed below:

Peterson, col. 15, lines 30-41:

Docket No. AUS920010638US1

Page 15 of 19

Atty Ref. No. IBM-1038

Sweitzer, et. al. - 09/996,131

## PATENT

Each category has a corresponding layer. These layers may contain reusable information, components, supply structures and entities. In this way a conceptual specification can be produced from the requirements analysis by using or reusing concepts from the conceptual layer. Similarly the functional specification is produced from the requirements specification by means of a mapping from the requirements specification using or reusing components from the functional layer (these components are functions and objects). This process also holds good for the interface and device layers. Of course, when no reusable component exists design must be performed at the lower levels.

Peterson, col. 15, lines 60-64:

Conceptual layer:

This layer contains, for example, abstract concepts such as topologies (network, star, ring) and design metaphors (copier interface, dishwasher interface, catalogue) and structures of tasks or tools (hierarchical, network).

Peterson, col. 26, lines 43-46:

Where a target service, for example, can be constructed from existing service features, so that what is to be produced for the customer is essentially a feature manager, the above method steps are sufficient in themselves.

The above sections of Peterson describe Peterson's system design process using Peterson's computerized design system. As described at length above, Peterson does not teach or suggest "topography components." The goals, subgoals, requirements, and specifications generated by Peterson's system might result in application components, but Peterson does not teach anything about the characteristics of such components. This is because Peterson is focused on a computerized system for aiding the requirements and design of computer system and is not directed towards the actual creation of such computer systems.

Consequently, as outlined above, claims 6, 13, and 19 are each allowable over the art of record for these reasons in addition to the fact that each of these claims depends upon an allowable independent claim.

Claims 7 and 20 include the limitation of "installing the selected topographical components on one or more client computer systems." The Office Action contends that Peterson teaches this limitation, citing col. 1, lines 37-43 and col. 9, lines 35-37 in support of the rejection. These sections of Peterson read as follows:

Peterson, col. 1, lines 37-43:

Docket No. AUS920010638US1

Page 16 of 19  
Sweitzer, et. al. - 09/996,131

Atty Ref. No. IBM-1038

**PATENT**

Prototyping requires the implementation of a simple version of the system under design (target design) which is then revised in a number of ensuing prototyping cycles, each cycle resulting in a prototype. The number of prototypes may vary from a few to several hundreds or even thousands, and may be followed by reimplementation of a final prototype in some delivery environment.

Peterson, col. 9, lines 35-37:

The customer or supplier organisation (sic) delivers the system to the user organisation (sic).

The first section (col. 1, lines 37-43) simply describes prototyping. Prototyping is often used during a system and analysis design lifecycle to test a system. Peterson does not teach or suggest that the “prototyping” is performed on a client computer system. Indeed, prototyping (being part of the system development process) typically does not involve installation on the computer system. The second section (col. 9, lines 35-37) simply states that a supplier delivers the system to a customer. Importantly, Peterson, in the cited sections or elsewhere, never teaches or suggests installing “selected” components on a client computer system. Additionally, Peterson does not even teach or suggest the creation of topographical components, so it would be impossible for Peterson to teach or suggest selecting such components and installing such selected components on a client computer system.

As outlined above, claims 7 and 20 are each allowable over the art of record for these reasons in addition to the fact that each of these claims depends upon an allowable independent claim.

Claims 21-23 each include limitations that include:

- selecting one of the topography neutral application components; and
- installing a first copy of the selected topography neutral application component on a first topology installation and a second copy of the selected topography neutral application component on a second topology installation, wherein the first and second topology installations are dissimilar topologies.

The Office Action rejected each of these claims contending that Peterson each of these limitations, citing col. 15, lines 60-64 and col. 16, lines 1-8 as allegedly teaching the “selecting” limitation and contending that Peterson teaches the “installing” limitation at col. 9, lines 35-37.

**PATENT**

Peterson, col. 15, lines 60-64:

Conceptual layer:

This layer contains, for example, abstract concepts such as topologies (network, star, ring) and design metaphors (copier interface, dishwasher interface, catalogue) and structures of tasks or tools (hierarchical, network).

Peterson, col. 16, lines 1-8:

The conceptual layer consists of subsystems and tool boxes. A subsystem can be regarded as a set of related tool boxes and a tool box as a set of related tools and entities (objects). A possible example is a service or system giving support to a sales organisation (sic). Some possible subsystems are the set of tools supporting various roles within the organisation (sic), other subsystems may be task related or goal related.

Peterson, col. 9, lines 35-37:

The customer or supplier organisation (sic) delivers the system to the user organisation (sic).

As previously discussed, the first two sections of Peterson set forth above are directed towards developing a "conceptual" layer of a system design. The last section simply states that a supplier delivers a system to a customer. Peterson does not teach or suggest the creation of "application components," nor does Peterson teach or suggest creating any "topography neutral" components. Finally, Applicants' claim limitations claim installing the application component on two dissimilar topographies. Nowhere does Peterson teach or suggest installation of anything on multiple topographies, let alone teach or suggest installing a common component on two dissimilar topographies.

Therefore, as discussed above, claims 21-23 are each allowable over Peterson for the reasons stated above as well as because each of these claims depends on an allowable independent claim.

**Conclusion**

As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims.

Docket No. AUS920010638US1

Page 18 of 19  
Sweitzer, et. al. - 09/996,131

Atty Ref. No. IBM-1038

## PATENT

Applicants respectfully request that the Examiner contact the Applicants' attorney listed below if the Examiner believes that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By

  
Joseph T. Van Leeuwen, Reg. No. 44,383  
Van Leeuwen & Van Leeuwen  
Attorneys for Applicant  
Telephone: (512) 301-6738  
Facsimile: (512) 301-6742

Docket No. AUS920010638US1

Page 19 of 19  
**Sweitzer, et. al. - 09/996,131**

Atty Ref. No. IBM-1038